DATA DELETION FOR A MULTI-TENANT ENVIRONMENT

BACKGROUND

[0001] Data storage systems write and read large amounts of data, and must also be able to delete data, both so that the storage system can recover and reuse memory, and so that the data that has been declared deleted is no longer accessible. Whether or not in a secure environment, data deletion prevents unauthorized access to data, and prevents erroneous use of data that has been superseded. For hard drives, secure deletion of data is often done by overwriting disk sectors multiple times, with constant or perhaps varying patterns. But, flash memory, and other forms of solid-state memory, suffer reduced lifespan upon such multiple overwrites. And, as amounts of data in storage grow, the amount of system resources and time that data deletion consumes grow as well, which can impact system performance. There is also the possibility that a client could request deletion of data, and later discover this was a mistake or the data is needed for reasons not known at the time of making the data deletion request, but find it is too late as the data deletion is in progress or completed. Even if a data deletion in progress can be stopped, data recovery will be compromised and incomplete, depending on how much of the data has already been deleted. These are some of the issues driving a need for improvements in data deletion mechanisms, which would be further beneficial if applicable in a multitenant environment.

BRIEF DESCRIPTION OF DRAWINGS

[0002] FIG. **1**A illustrates a first example system for data storage in accordance with some implementations.

[0003] FIG. **1**B illustrates a second example system for data storage in accordance with some implementations.

[0004] FIG. **1**C illustrates a third example system for data storage in accordance with some implementations.

[0005] FIG. **1**D illustrates a fourth example system for data storage in accordance with some implementations.

[0006] FIG. **2**A is a perspective view of a storage cluster with multiple storage nodes and internal storage coupled to each storage node to provide network attached storage, in accordance with some embodiments.

[0007] FIG. **2**B is a block diagram showing an interconnect switch coupling multiple storage nodes in accordance with some embodiments.

[0008] FIG. **2**C is a multiple level block diagram, showing contents of a storage node and contents of one of the non-volatile solid state storage units in accordance with some embodiments.

[0009] FIG. **2**D shows a storage server environment, which uses embodiments of the storage nodes and storage units of some previous figures in accordance with some embodiments.

[0010] FIG. **2**E is a blade hardware block diagram, showing a control plane, compute and storage planes, and authorities interacting with underlying physical resources, in accordance with some embodiments.

[0011] FIG. **2**F depicts elasticity software layers in blades of a storage cluster, in accordance with some embodiments.

[0012] FIG. **2**G depicts authorities and storage resources in blades of a storage cluster, in accordance with some embodiments.

[0013] FIG. **3**A sets forth a diagram of a storage system that is coupled for data communications with a cloud services provider in accordance with some embodiments of the present disclosure.

[0014] FIG. **3**B sets forth a diagram of a storage system in accordance with some embodiments of the present disclosure.

[0015] FIG. **3**C sets forth an example of a cloud-based storage system in accordance with some embodiments of the present disclosure.

[0016] FIG. **3**D illustrates an exemplary computing device that may be specifically configured to perform one or more of the processes described herein.

[0017] FIG. **4** depicts a storage system with a multitenant environment, storing data from multiple tenants, in encrypted form, using data management and key management.

[0018] FIG. **5** depicts a timeline and mechanism for deleting data, and undeleting data during a data recoverability time span, which can be used in embodiments of the storage system depicted in FIG. **4**.

[0019] FIG. **6** depicts a further mechanism for deleting data and undeleting data, which can be used in embodiments of the storage system depicted in FIG. **4**.

[0020] FIG. **7** is a flow diagram of a method for secure data deletion in a multitenant environment, which can be performed by embodiments of a storage system depicted in FIGS. **4**, **5** and **6**, and variations thereof.

[0021] FIG. **8** is a flow diagram of a method for secure data deletion and undeletion in a multitenant environment, which can be performed by embodiments of a storage system depicted in FIGS. **4** and **5**, and variations thereof.

[0022] FIG. **9** is a flow diagram of a method for secure data deletion and undeletion in a multitenant environment, which can be performed by embodiments of a storage system depicted in FIGS. **4** and **6**, and variations thereof.

DESCRIPTION OF EMBODIMENTS

[0023] Data storage systems that perform secure data deletion in a multitenant environment and are suitable for enterprises, storage providers and service providers, and others, are described herein. A large software development and hosting customer may have the need to guarantee to their customers that when a customer ends a contract the relevant data is fully deleted. Data deletion on flash memory does not work the same way as on disk, because a storage system cannot just overwrite the disk sectors multiple times without dramatically reducing the lifespan of the NAND flash memory cells and flash memory as a whole. A feasible way to truly delete data stored on flash memory is to encrypt before writing and then delete the encryption key. This makes the data wholly inaccessible—functionally the same as deletion. Garbage collection at a flash level will then re-use the memory space over time. For a multitenant environment, multiple keys could be used, e.g., one per tenant if a "tenant tagging" mechanism is available as in some embodiments described herein.

[0024] A storage system that uses global data reduction may make this approach difficult to impossible, and a storage system that only uses compression is a strong candidate for the data deletion mechanisms described herein. If the "tenant tag" is granular enough, this could even be part of a GDPR (General Data Protection Regulation, European Union) "Right To Be Forgotten" strategy.